

PCT/JP 2004/016264

日 本 国 特 許 庁
JAPAN PATENT OFFICE

04.11.2004

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 1 1 月 1 8 日
Date of Application:

REC'D 23 DEC 2004

WIPO

POT

出 願 番 号 特 願 2 0 0 3 - 3 8 7 3 5 0
Application Number:
[ST. 10/C]: [J P 2 0 0 3 - 3 8 7 3 5 0]

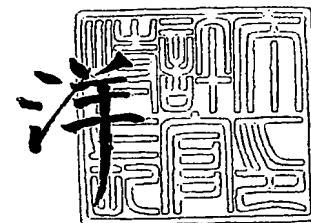
出 願 人 松下電器産業株式会社
Applicant(s):

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

2 0 0 4 年 1 2 月 1 3 日

特許庁長官
Commissioner,
Japan Patent Office

小 川



出証番号 出証特 2 0 0 4 - 3 1 1 3 5 8 1

【書類名】 特許願
【整理番号】 2037950004
【あて先】 特許庁長官殿
【国際特許分類】 G06F 12/08 310
【発明者】
 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
 【氏名】 岡林 はづき
【発明者】
 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
 【氏名】 中西 龍太
【発明者】
 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
 【氏名】 田中 哲也
【特許出願人】
 【識別番号】 000005821
 【氏名又は名称】 松下電器産業株式会社
【代理人】
 【識別番号】 100109210
 【弁理士】
 【氏名又は名称】 新居 広守
【手数料の表示】
 【予納台帳番号】 049515
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1
 【包括委任状番号】 0213583

【書類名】 特許請求の範囲**【請求項 1】**

ラインデータを保持するキャッシュエントリー毎に、当該キャッシュエントリーにキャッシングを終了してよいかどうかを示すキャッシング終了属性を付与する付与手段と、

キャッシングを終了してよいことを示すキャッシング終了属性が付与され、かつ書き込みされたことを示すダーティフラグがセットされているキャッシュエントリーを選択する選択手段と、

選択されたキャッシュエントリーのラインデータをキャッシュミスの発生と無関係にメモリへライトバックするライトバック手段と

を備えることを特徴とするキャッシュメモリ。

【請求項 2】

前記付与手段は、

プロセッサから指定されたアドレス範囲を保持する保持手段と、

保持手段に保持されたアドレス範囲に属するラインデータを保持するキャッシュエントリーを探索する探索手段と、

探索されたキャッシュエントリーに、キャッシングを終了してよいことを示すキャッシング終了属性を設定する設定手段と

を備えることを特徴とする請求項 1 記載のキャッシュメモリ。

【請求項 3】

前記探索手段は、

保持手段に保持されたアドレス範囲の先頭アドレスがラインデータの途中を指す場合、当該先頭アドレスを、前記アドレス範囲に含まれる先頭のラインを指すスタートラインアドレスに変換する第 1 変換手段と、

保持手段に保持されたアドレス範囲の末尾アドレスがラインデータの途中を指す場合、当該末尾アドレスを、前記アドレス範囲に含まれる末尾のラインを指すエンドラインアドレスに変換する第 2 変換手段と、

前記スタートラインアドレスからエンドラインアドレスまでの各ラインアドレスに対応するデータを保持するキャッシュエントリーがあるか否かを判定する判定手段とを備えることを特徴とする請求項 2 記載のキャッシュメモリ。

【請求項 4】

前記キャッシュメモリは、さらに、キャッシュミス発生時に、キャッシングを終了してよいことを示すキャッシング終了属性が付されたキャッシュエントリーをリプレース対象として選択するリプレース手段を備える

ことを特徴とする請求項 2 記載のキャッシュメモリ。

【請求項 5】

前記付与手段は、

プロセッサにおいてキャッシングを終了してよいことを示すキャッシング終了属性の付与とデータの書き込みとを命令内容とするストア命令が実行されたことを検出する命令検出手段と、

検出された当該命令によって書き込みがなされたキャッシュエントリーに、キャッシング終了属性を設定する設定手段と

を備えることを特徴とする請求項 1 記載のキャッシュメモリ。

【請求項 6】

前記ライトバック手段は、メモリバスが空きサイクルになっているとき、キャッシュエントリーのデータをメモリへライトバックする

ことを特徴とする請求項 1 記載のキャッシュメモリ。

【請求項 7】

前記キャッシュエントリーは、1 ラインを構成する複数のサブライン毎にダーティフラグを有し、

前記ライトバック手段は、選択手段に選択されたキャッシュエントリーについてダーテ

イなサブラインのみをライトバックする
ことを特徴とする請求項 1 記載のキャッシュメモリ。

【請求項 8】

キャッシュメモリの制御方法であって、
ラインデータを保持するキャッシュエントリー毎に、キャッシングを終了してよいかどうかを示すキャッシング終了属性を付与する付与ステップと、
キャッシングを終了してよいことを示すキャッシング終了属性が付与され、かつ書き込みされたことを示すダーティフラグがセットされているキャッシュエントリーを選択する選択ステップと、
選択されたキャッシュエントリーのラインデータをキャッシュミスの発生と無関係にメモリへライトバックするライトバックステップと
を有することを特徴とする制御方法。

【書類名】明細書**【発明の名称】キャッシュメモリ及びその制御方法****【技術分野】****【0001】**

本発明は、プロセッサのメモリアクセスを高速化するためのキャッシュメモリおよびその制御方法に関する。

【背景技術】**【0002】**

近年、キャッシュメモリは、メインメモリへのアクセス時間を短縮しプロセッサの処理能力の向上を図るために広く用いられている。

プロセッサからキャッシュメモリにデータが書き込まれた場合、メインメモリとキャッシュメモリとでデータの一貫性を確保するためにキャッシュメモリからメインメモリへ書き戻す必要がある。この書き戻しの制御方式には、ライトスルー方式およびライトバック方式がある。

【0003】

ライトスルー方式では、プロセッサからキャッシュメモリへのライト命令実行時にキャッシュメモリと主記憶との両方に書き込みを行う。このライトスルー方式によれば、キャッシュメモリにデータを書き込む毎にライトバックペナルティが発生するので、メモリライトに対してメモリアクセス時間を短縮する効果が薄い。

一方、ライトバック方式は、ライト命令実行時にキャッシュメモリのみに書き込みを行い、キャッシュミスによりダーティラインを新たな別のデータに置換する直前にキャッシュメモリからメインメモリにライトバックする。このライトバック方式は、メモリライト毎にライトバックペナルティが発生するわけではないが、メモリリード及びメモリライトにおいてキャッシュミス発生時にはロードペナルティに加えてライトバックペナルティを発生させるためキャッシュミス時のペナルティが増えることになる。ここでロードペナルティは、メインメモリからキャッシュメモリに新たなデータをロードすることによるペナルティをいう。ライトバックペナルティは、キャッシュメモリからメインメモリにデータを書き戻すことによるペナルティをいう。

【0004】

このようなキャッシュミスペナルティを低減する従来技術は、例えば、特許文献1等に開示されている。

この従来技術によれば、ライトバック方式のキャッシュメモリにおいて、キャッシュメモリからデータを追い出すためのメインメモリへの通常書き戻し処理とは別に、キャッシュメモリのデータをメインメモリに書き戻すだけのための書き戻し処理を行う。後者の処理は、先ずキャッシュメモリの一つのエントリーのデータがダーティであるか否かを判定し、ダーティであった場合はそのデータをメインメモリに書き戻す。そして、書き戻しが正常終了し、かつキャッシュメモリ上のデータが変化していないことを確認し、キャッシュ状態をクリーンとする。

【0005】

このようにして従来技術におけるキャッシュメモリは、キャッシュメモリからメインメモリへのトラフィックを減少させ、システム全体の性能向上を図っている。

【特許文献1】特開平6-309231号公報

【発明の開示】**【発明が解決しようとする課題】****【0006】**

しかしながら、上記従来技術によれば、キャッシュミスが発生する前にダーティなエントリーのデータをメモリにライトバックしても、そのライトバックの直後に当該エントリーにデータが書き込まれた場合に、却って効率が低下するという問題がある。

上記課題に鑑み本発明は、キャッシュミス時のライトバックペナルティを効率よく低減するキャッシュメモリを提供することを目的とする。

【課題を解決するための手段】

【0007】

上記目的を達成するため本発明のキャッシュメモリは、ラインデータを保持するキャッシュエントリー毎に、当該キャッシュエントリーにキャッシングを終了してよいかどうかを示すキャッシング終了属性を付与する付与手段と、キャッシングを終了してよいことを示すキャッシング終了属性が付与され、かつ書き込みされたことを示すダーティフラグがセットされているキャッシュエントリーを選択する選択手段と、選択されたキャッシュエントリーのラインデータをキャッシュミスの発生と無関係にメモリへライトバックするライトバック手段とを備える。

【0008】

この構成によれば、キャッシングを終了してよいキャッシュエントリーを、キャッシュミスが発生する前にライトバックするので、キャッシュミス時にはロードペナルティが発生するだけでライトバックペナルティの発生を低減することができる。これによりキャッシュメモリの効率を向上させ、アクセス速度を向上させることができる。

【0009】

ここで、前記付与手段は、プロセッサから指定されたアドレス範囲を保持する保持手段と、保持手段に保持されたアドレス範囲に属するラインデータを保持するキャッシュエントリーを探索する探索手段と、探索されたキャッシュエントリーに、キャッシングを終了してよいことを示すキャッシング終了属性を設定する設定手段とを備える構成としてもよい。

【0010】

ここで、前記探索手段は、保持手段に保持されたアドレス範囲の先頭アドレスがラインデータの途中を指す場合、当該先頭アドレスを、前記アドレス範囲に含まれる先頭のラインを指すスタートラインアドレスに変換する第1変換手段と、保持手段に保持されたアドレス範囲の末尾アドレスがラインデータの途中を指す場合、当該末尾アドレスを、前記アドレス範囲に含まれる末尾のラインを指すエンドラインアドレスに変換する第2変換手段と、前記スタートラインアドレスからエンドラインアドレスまでの各ラインアドレスに対応するデータを保持するキャッシュエントリーがあるか否かを判定する判定手段とを備える構成としてもよい。

【0011】

この構成によれば、プロセッサは、キャッシングを終了してよいことを示すキャッシング終了属性を設定すべきアドレス範囲として、キャッシュメモリのラインサイズやライン境界とは無関係に任意のアドレスを指定することができる。言い換えれば、プログラマやコンパイラは、ラインサイズ及びライン境界を意識する必要がなくなるので、ライトバックするためのラインアドレスを管理をしなくてもよく、プログラムの作成を容易にすることができる。

【0012】

ここで、前記キャッシュメモリは、さらに、キャッシュミス発生時に、キャッシングを終了してよいことを示すキャッシング終了属性が付されたキャッシュエントリーをリプレイス対象として選択する選択手段を備える構成としてもよい。

この構成によれば、アクセス頻度の低いデータを保持するキャッシュエントリーに、キャッシングを終了してよいことを示すキャッシング終了属性を設定することにより、当該キャッシュエントリーは真っ先にリプレイスされるので、アクセス頻度の低いデータがキャッシュメモリに居座ることによるキャッシュミスの誘発を低減することができる。

【0013】

ここで、前記付与手段は、プロセッサにおいてキャッシングを終了してよいことを示すキャッシング終了属性の付与とデータを書き込みとを命令内容とするストア命令が実行されたことを検出する命令検出手段と、当該命令によって書き込みがなされたキャッシュエントリーに、当該キャッシング終了属性を設定する設定手段とを備える構成としてもよい。

【0014】

この構成によれば、キャッシング終了属性の付与とデータを書き込みとを命令するストア命令により、キャッシングを終了してよいことを示すキャッシング終了を簡単に設定することができる。

【0015】

ここで、前記ライトバック手段は、メモリバスが空きサイクルになっているとき、キャッシュエントリーのデータをメモリへライトバックする構成としてもよい。

この構成によれば、空きサイクルを利用した効率のよいライトバックを行うことができる。

【0016】

ここで、前記キャッシュエントリーは、1ラインを構成する複数のサブライン毎にダーティフラグを有し、前記ライトバック手段は、選択手段に選択されたキャッシュエントリーについてダーティなサブラインのみをライトバックする構成としてもよい。

【0017】

この構成によれば、ダーティなサブラインのみをライトバックするので、ライトバックを効率よく行うことができる。

また、本発明のキャッシュメモリの制御方法についても上記と同様の手段、作用を有する。

【発明の効果】**【0018】**

本発明のキャッシュメモリによれば、以後に書き込みが行われないうキャッシュエントリーにキャッシングを終了してよいことを示すキャッシング終了属性を付与し、当該キャッシュエントリーを、キャッシュミスの発生とは無関係につまりキャッシュミスが発生する前にライトバックすることができる。その結果、キャッシュミス時にはロードペナルティが発生するだけでライトバックペナルティの発生を低減することができる。これによりキャッシュメモリの効率を向上させ、アクセス速度を向上させることができる。

【0019】

プロセッサは、ライトバックのためのラインアドレスの管理をする必要がなく、キャッシュメモリのラインサイズ及びライン境界とは無関係に任意のアドレス範囲を指定することができる。

【発明を実施するための最良の形態】**【0020】****(実施の形態1)****<全体構成>**

図1は、本発明の実施の形態1におけるプロセッサ1、キャッシュメモリ3、メモリ2を含むシステムの概略構成を示すブロック図である。同図のように、本発明のキャッシュメモリ3は、プロセッサ1およびメモリ2を有するシステムに備えられる。

【0021】

キャッシュメモリ3は、キャッシュの単位データ（ラインデータと呼ぶ）を保持するキャッシュエントリー毎に、当該キャッシュエントリーにキャッシングを終了してよいかどうかを示すキャッシング終了属性を保持し、キャッシュミスの発生の有無に関わらず、キャッシングを終了してよいことを示すキャッシング終了属性を有しかつプロセッサから書き込みされたことを示すダーティフラグがセットされているキャッシュエントリーを選択し、選択されたキャッシュエントリーのデータをメモリへライトバックするよう構成されている。ここで、キャッシングを終了してよいことを示すキャッシング終了属性は、例えば、以後に書き込みが行われないうキャッシュエントリーや、以降に読み書きされないであろうキャッシュエントリー等に付与される。

【0022】**<キャッシュメモリの構成>**

以下、キャッシュメモリ3の具体例として、4ウェイ・セット・アソシエイティブ方式

のキャッシュメモリに本発明を適用した場合の構成について説明する。

【0023】

図2は、キャッシュメモリ3の構成例を示すブロック図である。同図のように、キャッシュメモリ3は、アドレスレジスタ20、メモリI/F21、デコーダ30、4つのウェイ31a~31d（以下ウェイ0~3と略す）、4つの比較器32a~32d、4つのアンド回路33a~33d、オア回路34、セクタ35、36、デマルチプレクサ37、制御部38を備える。

【0024】

アドレスレジスタ20は、メモリ2へのアクセスアドレスを保持するレジスタである。このアクセスアドレスは32ビットであるものとする。同図に示すように、アクセスアドレスは、最上位ビットから順に、21ビットのタグアドレス、4ビットのセットインデックス（図中のSI）、5ビットのワードインデックス（図中のWI）を含む。ここで、タグアドレスはウェイにマッピングされるメモリ中の領域（そのサイズはセット数×ブロックである）を指す。この領域のサイズは、タグアドレスよりも下位のアドレスビット（A10~A0）で定まるサイズつまり2kバイトであり、1つのウェイのサイズでもある。セットインデックス（SI）はウェイ0~3に跨る複数セットの1つを指す。このセット数は、セットインデックスが4ビットなので16セットある。タグアドレスおよびセットインデックスで特定されるキャッシュエントリは、リプレース単位であり、キャッシュメモリに格納されている場合はラインデータ又はラインと呼ばれる。ラインデータのサイズは、セットインデックスよりも下位のアドレスビットで定まるサイズつまり128バイトである。1ワードを4バイトとすると、1ラインデータは32ワードである。ワードインデックス（WI）は、ラインデータを構成する複数ワード中の1ワードを指す。アドレスレジスタ20中の最下位2ビット（A1、A0）は、ワードアクセス時には無視される。

【0025】

メモリI/F21は、キャッシュメモリ3からメモリ2へのデータのライトバックや、メモリ2からキャッシュメモリ3へのデータのロード等、キャッシュメモリ3からメモリ2をアクセスするためのI/Fである。

デコーダ30は、セットインデックスの4ビットをデコードし、4つのウェイ0~3に跨る16セット中の1つを選択する。

【0026】

4つのウェイ0~3は、同じ構成を有数する4つのウェイであり、4×2kバイトの容量を有する。各ウェイは、16個のキャッシュエントリを有する。

図3に1つのキャッシュエントリにおける詳細なビット構成を示す。同図のように、1つのキャッシュエントリは、バリッドフラグV0~V3、21ビットのタグ、128バイトのラインデータ、キャッシング終了属性（図中のCフラグ）、ダーティフラグD0~D3を有する。

【0027】

タグは21ビットのタグアドレスのコピーである。

ラインデータは、タグアドレスおよびセットインデックスにより特定されるブロック中の128バイトデータのコピーであり、32バイトの4つのサブラインからなる。

バリッドフラグV0~V3は、4つのサブラインに対応し、サブラインが有効か否かを示す。

【0028】

キャッシング終了属性（クリーニングフラグC）は、キャッシングを終了してよいかどうかを示す。例えば、当該キャッシュエントリに以降に書き込みが行われないうかがを意味する。C=0は、以降に書き込みがなされる可能性があることを意味する。C=1は、以降に書き込みがなされないことを意味し、ダーティであればクリーニング（ライトバック）すべきであることを意味する。

【0029】

ダーティフラグ D0～D3 は、4つのサブラインに対応し、そのサブラインにプロセッサから書き込みがあったか否か、つまりサブライン中にキャッシュされたデータが存在するが書き込みによりメモリ中のデータと異なるためメモリに書き戻すことが必要か否かを示す。

比較器 32a は、アドレスレジスタ 20 中のタグアドレスと、セットインデックスにより選択されたセットに含まれる 4つのタグ中のウェイ 0 のタグとが一致するか否かを比較する。比較器 32b～32c についても、ウェイ 31b～31d に対応すること以外は同様である。

【0030】

アンド回路 33a は、バリッドフラグと比較器 32a の比較結果とが一致するか否かを比較する。この比較結果を h0 とする。比較結果 h0 が 1 である場合は、アドレスレジスタ 20 中のタグアドレスおよびセットインデックスに対応するラインデータが存在すること、つまりウェイ 0 においてヒットしたことを意味する。比較結果 h0 が 0 である場合は、ミスヒットしたことを意味する。アンド回路 33b～33d についても、ウェイ 31b～31d に対応すること以外は同様である。その比較結果 h1～h3 は、ウェイ 1～3 でヒットしたかミスしたかを意味する。

【0031】

オア回路 34 は、比較結果 h0～h3 のオアをとる。このオアの結果を hit とする。hit は、キャッシュメモリにヒットしたか否かを示す。

セレクトア 35 は、選択されたセットにおけるウェイ 0～3 のラインデータのうち、ヒットしたウェイのラインデータを選択する。

セレクトア 36 は、セレクトア 35 により選択された 32 ワードのラインデータのうち、ワードインデックスに示される 1 ワードを選択する。

【0032】

デマルチプレクサ 37 は、キャッシュエントリにデータを書き込む際に、ウェイ 0～3 の 1 つに書き込みデータを出力する。この書き込みデータはワード単位でよい。

制御部 38 は、キャッシュメモリ 3 の全体の制御を行う。特に、C フラグの設定と C フラグに従うクリーニング（ライトバック）とを行う。

【0033】

<制御部の構成>

図 4 は、制御部 38 の構成を示すブロック図である。同図のように、制御部 38 は、クリーニング処理部 39 と C フラグ設定部 40 とを含む。

【0034】

クリーニング処理部 39 は、C=1 が設定されているキャッシュエントリを探索し、当該キャッシュエントリがダーティであればライトバックする。

C フラグ設定部 40 は、プロセッサ 1 からのコマンドに応じてクリーニングフラグ C を設定する。プロセッサ 1 は、もはや書き込みをしないキャッシュエントリについてクリーニングフラグの設定を指示するコマンドをキャッシュメモリ 3 に対して発行する。

【0035】

<C フラグ設定部の構成>

図 5 は、C フラグ設定部 40 の構成例を示すブロック図である。同図のように C フラグ設定部 40 は、コマンドレジスタ 401、スタートアドレスレジスタ 402、サイズレジスタ 403、加算器 404、スタートアライナ 405、エンドアライナ 406、フラグ書換部 407 を備える。

【0036】

コマンドレジスタ 401 は、プロセッサ 1 から直接アクセス可能なレジスタであり、プロセッサ 1 により書き込まれた C フラグ設定コマンドを保持する。図 6 (c) に、コマンドレジスタ 401 にコマンドを書き込む命令の一例を示す。この命令は、通常の転送命令 (mov 命令) であり、ソースオペランドとしてコマンドを、デスティネーションオペランドとしてコマンドレジスタ (CR) 401 を指定している。図 6 (d) に、コマンドの

一例を示す。このコマンドは、Cフラグ設定コマンドを示す特定のコードである。Cフラグ設定コマンドは、スタートアドレスレジスタ402に保持されたスタートアドレスからサイズレジスタ403に保持されたサイズのアドレス範囲に対応するデータを保持するキャッシュエントリーに対して、Cフラグを設定することを指示するコマンドである。

【0037】

スタートアドレスレジスタ402は、プロセッサ1から直接アクセス可能なレジスタであり、プロセッサ1により書き込まれたスタートアドレスを保持する。このスタートアドレスはCフラグを設定すべきアドレス範囲の開始位置を示す。図6(a)に、スタートアドレスレジスタ402にスタートアドレスを書き込む命令の一例を示す。この命令も、図6(c)と同様に通常の転送命令(mov命令)である。

【0038】

サイズレジスタ403は、プロセッサ1から直接アクセス可能なレジスタであり、プロセッサ1により書き込まれたサイズを保持する。このサイズは、スタートアドレスからのアドレス範囲を示す。図6(b)に、サイズレジスタ403にサイズを書き込む命令の一例を示す。この命令も、図6(c)と同様に通常の転送命令(mov命令)である。なお、サイズの単位は、バイト数であっても、ライン数(キャッシュエントリー数)であってもよく、予め定められた単位であればよい。

【0039】

加算器404は、スタートアドレスレジスタ402に保持されたスタートアドレスとサイズレジスタ403に保持されたサイズとを加算する。加算結果は、アドレス範囲の終了位置を指すエンドアドレスである。加算器404は、サイズがバイト数指定の場合はバイトアドレスとして加算し、サイズがライン数指定の場合はラインアドレスとして加算すればよい。

【0040】

スタートアライナ405は、スタートアドレスをライン境界の位置に調整する。この調整によりプロセッサ1はラインサイズ及びライン境界とは無関係に任意のアドレスをスタートアドレスとして指定することができる。

エンドアライナ406は、エンドアドレスをライン境界の位置に調整する。この調整によりプロセッサ1はラインサイズ及びライン境界とは無関係に任意の大きさを上記サイズとして指定することができる。

【0041】

図7に、スタートアライナ405及びエンドアライナ406の説明図を示す。同図において、プロセッサ1から指定されたスタートアドレスはラインNの途中の任意の位置を指す。スタートアライナ405は、次のライン(N+1)の先頭を指すよう調整し、調整後のアドレスをアラインスタートアドレスとして出力する。アラインスタートアドレスが指すラインをスタートラインと呼ぶ。

【0042】

また、エンドアドレスはラインMの途中の任意の位置を指す。エンドアライナ406は、直前のライン(M-1)の先頭を指すよう調整し、調整後のアドレスをアラインエンドアドレスとして出力する。アラインエンドアドレスが指すラインをエンドラインと呼ぶ。

この場合、スタートライン(ライン(N+1))からエンドライン(ライン(M-1))までの各ライン(キャッシュエントリー)にCフラグが設定されることになる。このように、スタートアライナ405及びエンドアライナ406がプロセッサ1から指定されたスタートアドレスからエンドアドレスまでのアドレス範囲よりも内側にアラインしているのは、ラインNとラインMの外側の部分にはプロセッサ1から書き込みが発生する可能性があるからである。

【0043】

フラグ書換部407は、アラインスタートアドレスが指すラインからアラインエンドアドレスが指すラインまで(図7の例ではライン(N+1)からライン(M-1)まで)、キャッシュメモリ3にエントリーされていればCフラグを1に設定する。

【0044】

<Cフラグ設定処理>

図8は、フラグ書換部407におけるCフラグ設定処理の一例を示すフローチャートである。

【0045】

フラグ書換部407は、コマンドレジスタ401にCフラグ設定コマンドが保持されている場合、スタートラインからエンドラインまでの各ラインアドレスを順に出力しながらループ1の処理(S82~S86)を行う。フラグ書換部407は、各ラインについて同じ処理を行うので、ここでは1ライン分の処理について説明する。

すなわち、フラグ書換部407は、キャッシュメモリ3がプロセッサ1からアクセスされていない間に、ラインアドレスをアドレスレジスタ20に出力し(S83)、アドレスレジスタ20のタグアドレスとキャッシュエントリーのタグとを比較器32a~32dに比較させ、ヒットするかどうかを判定する(S84)。さらにフラグ書換部407は、ヒットした場合には、ヒットしたキャッシュエントリーに対してCフラグを1にセットし(S85)、ミスヒットした場合には、キャッシュメモリにエントリーされていないのでなにもしない。

これにより、スタートラインからエンドラインまでの各ラインについて、キャッシュメモリ3にエントリーされている場合には、Cフラグに1が設定される。

【0046】

<クリーニング処理>

図9は、クリーニング処理部39におけるクリーニング処理の一例を示すフローチャートである。

【0047】

同図のように、クリーニング処理部39は、ループ1の処理(S900~S913)において、セットインデックス(SI)0~15を順に指定する(S901)ことにより、16個の全てのセットに対してループ2の処理を行う。ループ2の処理(S900~S913)において、クリーニング処理部39は、セット内のウェイ0~3のCフラグを読み出す(S903)ことにより、C=1のキャッシュエントリーを探索する(S904)。ループ3の処理(S905~910)において、クリーニング処理部39は、C=1のキャッシュエントリーに対して、サブライン単位のダーティフラグを読み出し(S906)、ダーティであれば(S907)、そのサブラインのデータをメモリ2に書き戻し(S908)、当該ダーティフラグを0にリセットする(S909)。このサブラインデータの書き戻しにおいて、クリーニング処理部39は、ループ4の処理(S920~S923)のように、空きサイクルにおいて(S920)、1ワードずつ書き戻す(S922)。

このように、クリーニング処理部39は、全てのキャッシュエントリーのCフラグを順にチェックして、C=1のキャッシュエントリーを探索し、ダーティであればキャッシュメモリ3からメモリ2に書き戻す。

【0048】

以上説明してきたように、本実施の形態におけるキャッシュメモリによれば、プロセッサ1から指定されたアドレス範囲に属するキャッシュエントリーに対して、キャッシングを終了してよいかどうかを示すキャッシング終了属性(Cフラグ)を付与し、これ以上書き込みされないキャッシュエントリーを、キャッシュミスが発生する前にライトバックするので、キャッシュミス時にはロードペナルティが発生するだけでライトバックペナルティの発生を低減することができる。これによりキャッシュメモリの効率を向上させ、アクセス速度を向上させることができる。

【0049】

しかも、プロセッサ1は、キャッシング終了属性を設定すべきアドレス範囲として、キャッシュメモリのラインサイズやライン境界とは無関係に任意のアドレスを指定することができる。これにより、プログラマやコンパイラは、ラインサイズ及びライン境界を意識する必要がなくなるので、クリーニングするためのキャッシュアドレス管理をしなくても

よく、プログラムの作成を容易にすることができる。

【0050】

さらに、クリーニング処理において、サブライン単位にダーティなサブランのみをライトバックするので、ライン単位でライトバックする場合と比較してライトバックを高速化することができる。

【0051】

<変形例>

なお、本発明のキャッシュメモリは、上記の実施形態の構成に限るものではなく、種々の変形が可能である。以下、変形例のいくつかについて説明する。

(1) 図5において、Cフラグ設定部40は、サイズレジスタ403及び加算器404の代わりにエンドアドレスレジスタを備える構成としてもよい。この場合、エンドアドレスレジスタは、プロセッサ1から書き込まれたエンドアドレスを保持する。

(2) プロセッサ1が、キャッシング終了属性を付与しながらデータを書き込むストア命令を実行し、制御部38は、さらに、キャッシング終了属性を付与しながらデータを書き込むストア命令を検出する命令検出部と、当該ストア命令による書き込みの際にC=1に設定するフラグ設定部とを備える構成としてもよい。

(3) 図6(a)(b)(c)に示した各命令は、コンパイラによりプログラム中に挿入してもよい。その際、コンパイラは、例えば配列データの書き込みや、圧縮動画データをデコードする際のブロックデータの書き込み等、これ以上書き込みをしないプログラム位置に、上記各命令を挿入するようにすればよい。

(4) キャッシュエントリーにCフラグを保持させない構成としてもよい。言い換えれば、図8に示したCフラグ設定処理と図9に示したクリーニング処理とを同時に行う構成としてもよい。この場合、図8におけるS85において、Cフラグをキャッシュエントリーに設定する代わりに、図9におけるループ3(S905~S910)を実行する構成とすればよい。

(5) 上記実施の形態では、4ウェイ・セット・アソシエイティブのキャッシュメモリを例に説明したが、ウェイ数は、8ウェイでも16ウェイでもよい。また、上記実施の形態では、セット数が16である例を説明したが、セット数はいくつでもよい。

(6) 上記実施の形態では、セット・アソシエイティブのキャッシュメモリを例に説明したが、フル・アソシエイティブ方式のキャッシュメモリであってもよい。

【0052】

(実施の形態2)

実施の形態1では、これ以上書き込みをするかしないかを示すCフラグを用いる構成について説明した。本実施の形態では、Cフラグの代わりに、これ以上使用するかしないか(書き込み及び読み出しをするかどうか)を示すW(ウィーク)フラグを用いる構成について説明する。

【0053】

<キャッシュメモリの構成>

図10は、本発明の実施の形態2におけるキャッシュメモリの構成を示すブロック図である。同図のキャッシュメモリは、図2の構成と比較して、ウェイ31a~31dの代わりにウェイ131a~131dを備える点と、制御部38の代わりに制御部138を備える点とが異なっている。以下、同じ点は説明を省略して、異なる点を中心に説明する。

【0054】

ウェイ131aは、ウェイ31aと比べて、各キャッシュエントリー中に、Cフラグが削除されている点と、Wフラグ及びUフラグが追加されている点が異なる。ウェイ131b~131dも同様である。

図11に、キャッシュエントリーのビット構成を示す。1つのキャッシュエントリーは、バリッドフラグV0~V3、21ビットのタグ、128バイトのラインデータ、ウィークフラグW、使用フラグU及びダーティフラグD0~D3を保持する。

【0055】

このうち、ウィークフラグWは、プロセッサからのアクセスに関しては、これ以上使用するか否かを意味し、キャッシュメモリにおけるリプレース制御に関しては、他のキャッシュエントリよりも先に追い出してもよい最弱のリプレース対象を意味する。このように、ウィークフラグWは二つの意味を有することから、クリーニング処理とリプレース処理との2つの処理で参照される。

【0056】

使用フラグUは、そのキャッシュエントリにアクセスがあったか否かを示し、LRU方式におけるミスヒットによるリプレースに際して4つのウェイのキャッシュエントリ間におけるアクセス順序データの代わりに用いられる。より正確には、使用フラグUの1は、アクセスがあったことを、0はないことを意味する。ただし、1つのセット内の4つウェイの使用フラグが全て1になれば、0にリセットされる。別言すれば、使用フラグUは、アクセスされた時期が古いか新しいか2つの相対的な状態を示す。つまり、使用フラグUが1のキャッシュエントリは、使用フラグが0のキャッシュエントリよりも新しくアクセスされたことを意味する。

【0057】

制御部138は、制御部38と比べて、CフラグではなくWフラグを設定する点と、LRU方式におけるアクセス順序情報の代わりに使用フラグUを用いる点とが異なる。

【0058】

<制御部の構成>

図12は、制御部138の構成を示すブロック図である。同図の制御部138は、制御部38と比較して、クリーニング処理部39とCフラグ設定部40との代わりにクリーニング処理部139とWフラグ設定部140とを備える点と、リプレース部41が追加された点とが異なる。

【0059】

クリーニング処理部139は、クリーニング処理部39と比べて、CフラグではなくWフラグを参照する点が異なっている。その結果、W=1でかつダーティなキャッシュラインをライトバックする。

Wフラグ設定部140は、プロセッサ1からのコマンドに応じてウィークフラグWを設定する。プロセッサ1は、もはや使用（書き込み及び読み出し）をしないキャッシュエントリについてウィークフラグの設定を指示するコマンドをキャッシュメモリ3に対して発行する。W=1のキャッシュエントリは、ダーティであればクリーニング処理の対象となり、また、キャッシュミス時には真つ先にリプレース対象となる。

【0060】

リプレース部41は、使用フラグUをアクセス順序とする擬似的なLRU方式によりキャッシュミス時にリプレース処理と、キャッシュメモリがアクセスされたとき使用フラグUの更新処理を行う。リプレース処理に際してW=1のキャッシュエントリは真つ先にリプレース対象として選択される。

【0061】

<使用フラグUの説明>

図13は、リプレース部41による使用フラグの更新例を示す。同図の上段、中断、下段は、ウェイ0～3に跨るセットNを構成する4つのキャッシュエントリを示している。4つのキャッシュエントリ右端の1又は0は、それぞれ使用フラグの値である。この4つの使用フラグUをU0～U3と記す。

【0062】

同図上段では(U0～U3)=(1, 0, 1, 0)であるので、ウェイ0、2のキャッシュエントリはアクセスがあったことを、ウェイ1、3のキャッシュエントリはアクセスがないことを意味する。

この状態で、メモリアクセスがセットN内のウェイ1のキャッシュエントリにヒットした場合、同図中段に示すように、(U0～U3)=(1, 1, 1, 0)に更新される。つまり、実線に示すようにウェイ1の使用フラグU1が0から1に更新される。

【0063】

さらに、同図中段の状態、メモリアクセスがセットN内のウェイ3のキャッシュエントリーにヒットした場合、同図下断に示すように、 $(U0 \sim U3) = (0, 0, 0, 1)$ に更新される。つまり、実線に示すようにウェイ3の使用フラグU1が0から1に更新される。加えて、破線に示すようにウェイ3以外の使用フラグU0～U2が1から0に更新される。これにより、ウェイ3のキャッシュエントリーが、ウェイ0～2の各キャッシュエントリーよりも新しくアクセスされたことを意味することになる。

【0064】

リプレース部41は、キャッシュミス時に $W=1$ のキャッシュエントリーが存在しなければ、使用フラグに基づいてリプレース対象のキャッシュエントリーを決定してリプレースを行う。例えば、リプレース部41は、図5上段では、ウェイ1とウェイ3の何れかをリプレース対象と決定し、図5中段ではウェイ3をリプレース対象と決定し、図5下段ではウェイ0～2の何れかをリプレース対象と決定する。

＜ウィークフラグWの説明＞

図14(a)ウィークフラグが存在しないと仮定した場合の比較例であり、キャッシュエントリーがリプレースされる様子を示す図である。同図においても、図13と同様にウェイ0～3に跨るセットNを構成する4つのキャッシュエントリーを示している。4つのキャッシュエントリー右端の1又は0は、それぞれ使用フラグの値である。また、データEのみアクセス頻度の低いデータを、データA、B、C、Dはアクセス頻度の高いデータとする。

【0065】

同図(a)の第1段目の状態で、プロセッサ1がデータEにアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、例えば、 $U=0$ のキャッシュエントリーの中からアクセス頻度の高いデータCのキャッシュエントリーがアクセス頻度の低いデータEにリプレースされ、第2段目の状態となる。

第2段目の状態で、プロセッサ1がデータCにアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、 $U=0$ のキャッシュエントリーであるアクセス頻度の高いデータDのキャッシュエントリーがアクセス頻度の高いデータCにリプレースされ、第3段目の状態となる。

【0066】

第3段目の状態で、プロセッサ1がデータDにアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、例えば、アクセス頻度の高いデータCのキャッシュエントリーがアクセス頻度の高いデータDにリプレースされ、第3段目の状態となる。

同様に、第4段目でも、使用頻度の低いデータEはリプレース対象として選択されないで、キャッシュメモリーに残っている。

【0067】

第5段目の状態で、使用頻度の低いデータEは最も古い($U=0$)であることから、リプレース対象として選択されて、追い出される。

このように、擬似LRU方式において(通常のLRU方式においても)、アクセス頻度の低いデータEによって、4ウェイの場合は最悪4回のキャッシュミスを誘発する場合がある。

【0068】

図14(b)は、リプレース処理におけるウィークフラグWの役割を示す説明図である。

同図(b)の第1段目の状態(同図(a)の第1段目と同じ)で、プロセッサ1がデータEにアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、例えば、 $U=0$ のキャッシュエントリーの中からアクセス頻度の高いデータCのキャッシュエントリーがアクセス頻度の低いデータEにリプレースされる。このとき、プロセッサ1は、データEのキャッシュエントリーにウィークフラグWを1に設定するものとする。これにより、次のキャッシュミス時にデータEのキャッシュエントリーが真っ先に追い出され、

第2段目の状態となる。

【0069】

第2段目の状態で、プロセッサ1がデータCにアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、 $W=1$ のキャッシュエントリーであるアクセス頻度の低いデータEのキャッシュエントリーがリプレース対象として選択され、アクセス頻度の高いデータCにリプレースされ、第3段目の状態となる。

このように、ウィークフラグWを設けることにより、アクセス頻度の低いデータによるキャッシュミスの誘発を低減することができる。

【0070】

<Wフラグ設定処理>

図15は、Wフラグ設定部140におけるWフラグ設定処理を示すフローチャートである。Wフラグ設定部140の構成は、図5に示したCフラグ設定部と同様であり、また、プロセッサ1からのWフラグ設定コマンドも、図6(a)～(d)と同様である。ただし、プロセッサ1は、これ以上使用しない(書き込みも読み出しもしない)データのアドレス範囲を、Wフラグ設定コマンドとともに指定する。

【0071】

図15は、図8と比べて、ステップS85の代わりにステップS185を有する点が異なっている。これ以外は、図8のCフラグ設定処理と同じであるので、説明を省略する。

ステップS185において、Wフラグ設定部140は、S84によってヒットしたキャッシュエントリーに $W=1$ を設定する。

【0072】

<クリーニング処理>

図16は、クリーニング処理部139におけるクリーニング処理を示すフローチャートである。

同図は、図9と比較して、S903、S904、S911の代わりにS903a、S904a、S911aを有する点が異なっている。S903a、S904a、S911aは、何れもCフラグをWフラグと読み替える点のみが異なり、それ以外は同じ処理内容であるので説明を省略する。

【0073】

<Uフラグ更新処理>

図17は、リプレース部41におけるUフラグ更新処理を示すフローチャートである。同図では、バリッドフラグが0(無効)であるキャッシュエントリーの使用フラグUは0に初期化されているものとする。

【0074】

同図において、リプレース部41は、キャッシュヒットしたとき(ステップS61)、セットインデックスにより選択されたセットにおけるヒットしたウェイの使用フラグUを1にセットし(ステップS62)、そのセット内の他のウェイの使用フラグUを読み出し(ステップS63)、読み出した使用フラグUが全て1であるか否かを判定し(ステップS64)、全て1でなければ終了し、全て1であれば他のウェイの全ての使用フラグUを0にリセットする(ステップS65)。

このようにしてリプレース部41は、図13、図14(a)(b)に示した更新例のように、使用フラグUを更新する。

【0075】

<リプレース処理>

図18は、リプレース部41におけるリプレース処理を示すフローチャートである。同図においてリプレース部41は、メモリアクセスがミスしたとき(ステップS91)、セットインデックスにより選択されたセットにおける、4つウェイの使用フラグU及びウィークフラグWを読み出し(ステップS92)、 $W=1$ のウェイが存在するか否かを判定する(ステップS93)。 $W=1$ のウェイが存在しないと判定された場合、 $U=0$ のウェイを1つ選択する(ステップS94)。このとき、使用フラグUが0になっているウェイが

複数存在する場合は、リプレース部 41 はランダムに 1 つを選択する。また、 $W=1$ のウェイが存在すると判定された場合、U フラグの値に関わらず $W=1$ のウェイを 1 つ選択する (ステップ S95)。このとき、ウィークフラグ W が 1 になっているウェイが複数存在する場合は、リプレース部 41 はランダムに 1 つを選択する。

【0076】

さらに、リプレース部 41 は、当該セットにおける選択されたウェイのキャッシュエントリーを対象にリプレースし (ステップ S96)、リプレース後に当該キャッシュエントリーの使用フラグ U を 1 に、ウィークフラグ W を 0 初期化する (ステップ S97)。なお、このときバリッドフラグ V 、ダーティフラグ D は、それぞれ 1、0 に初期化される。

このように、 $W=1$ のウェイが存在しない場合、リプレース対象は、使用フラグ U が 0 のキャッシュエントリーの中から 1 つ選択される。

【0077】

また、 $W=1$ のウェイが存在する場合、リプレース対象は、使用フラグ U が 0 であると 1 であることを問わず、 $W=1$ のウェイのキャッシュエントリーから 1 つ選択される。これにより図 14 (a) (b) に示したように、アクセス頻度の低いデータがキャッシュメモリに残ることによるキャッシュミスの誘発を低減することができる。

以上説明してきたように、本実施の形態におけるキャッシュメモリによれば、ウィークフラグ $W=1$ のラインを、プロセッサからこれ以上書き込みがなされないラインとして、クリーニングすることにより、キャッシュミス時のライトバックペナルティを低減することができる。

【0078】

また、これ以上使用されないキャッシュエントリーに $W=1$ が設定され、 $W=1$ のキャッシュエントリーが真っ先にリプレース対象として選択されるので、アクセス頻度の低いデータがキャッシュメモリに残ることによるキャッシュミスの誘発を低減することができる。

また、従来の LRU 方式におけるアクセス順序を示すデータの代わりに 1 ビットの使用フラグを用いる擬似 LRU 方式を採用することにより、アクセス順序データとして 1 ビットのフラグでよいので、アクセス順序データのデータ量が少ないこと及び更新が簡単であることからハードウェア規模を小さくすることができる。

【0079】

<変形例>

なお、本発明のキャッシュメモリは、上記の実施形態の構成に限るものではなく、種々の変形が可能である。以下、変形例のいくつかについて説明する。

(1) 使用フラグ U を用いる擬似 LRU の代わりに、4 つのウェイのアクセス順序を示す順序データをキャッシュエントリー毎に保持及び更新して、従来通りの LRU 方式でリプレース対象を選択する構成としてもよい。この場合も、 $W=1$ のキャッシュエントリーを、アクセス順序に関わらず、真っ先にリプレース対象として選択するようにすればよい。

(2) 実施の形態 1 に示した C フラグと、本実施の形態における W フラグの両方設ける構成としてもよい。この場合、C フラグはクリーニング処理で用いて、 W フラグはリプレース対象の選択のみに用いるようにすればよい。こうすれば、C フラグによって、読み出しされる可能性があつたこれ以上書き込みされないキャッシュエントリーは、クリーニングされた後もキャッシュメモリに残すことができる。これ以上読み出しも書き込みもなされないキャッシュエントリーは、クリーニングされかつリプレース対象として真っ先に選択される。C フラグと W フラグの併用によって、きめ細かい制御を行うことができる。

(3) 上記実施の形態では、サブラインのサイズをラインのサイズの $1/4$ としているが、 $1/2$ 、 $1/8$ 、 $1/16$ 等他のサイズでもよい。その場合、各キャッシュエントリーは、サブラインと同数のバリッドフラグおよびダーティフラグをそれぞれ保持すればよい。

【産業上の利用可能性】

【0080】

本発明は、メモリアクセスを高速化するためのキャッシュメモリに適しており、例えば、オンチップキャッシュメモリ、オフチップキャッシュメモリ、データキャッシュメモリ、命令キャッシュメモリ等に適している。

【図面の簡単な説明】

【0081】

【図1】本発明の実施の形態1におけるプロセッサ、キャッシュメモリ、メモリを含むシステムの概略構成を示すブロック図である。

【図2】キャッシュメモリの構成例を示すブロック図である。

【図3】キャッシュエントリーの詳細なビット構成を示す図である。

【図4】制御部の構成を示すブロック図である。

【図5】Cフラグ設定部の構成例を示すブロック図である。

【図6】(a) スタートアドレスレジスタにスタートアドレスを書き込む命令の一例を示す。(b) サイズレジスタにサイズを書き込む命令の一例を示す。(c) コマンドレジスタにコマンドを書き込む命令の一例を示す。(d) コマンドの一例を示す。

【図7】スタートアライナ及びエンドアライナの説明図を示す。

【図8】フラグ書換部におけるCフラグ設定処理の一例を示すフローチャートである。

【図9】クリーニング処理部におけるクリーニング処理の一例を示すフローチャートである。

【図10】本発明の実施の形態2におけるキャッシュメモリの構成を示すブロック図である。

【図11】キャッシュエントリーのビット構成を示す図である。

【図12】制御部の構成を示すブロック図である。

【図13】リプレース部による使用フラグの更新例を示す。

【図14】(a) ウィークフラグが存在しない場合にキャッシュエントリーがリプレースされる様子を示す図である。(b) リプレース処理におけるウィークフラグWの役割を示す説明図である。

【図15】Wフラグ設定部におけるWフラグ設定処理を示すフローチャートである。

【図16】クリーニング処理部におけるクリーニング処理を示すフローチャートである。

【図17】リプレース部におけるUフラグ更新処理を示すフローチャートである。

【図18】リプレース部におけるリプレース処理を示すフローチャートである。

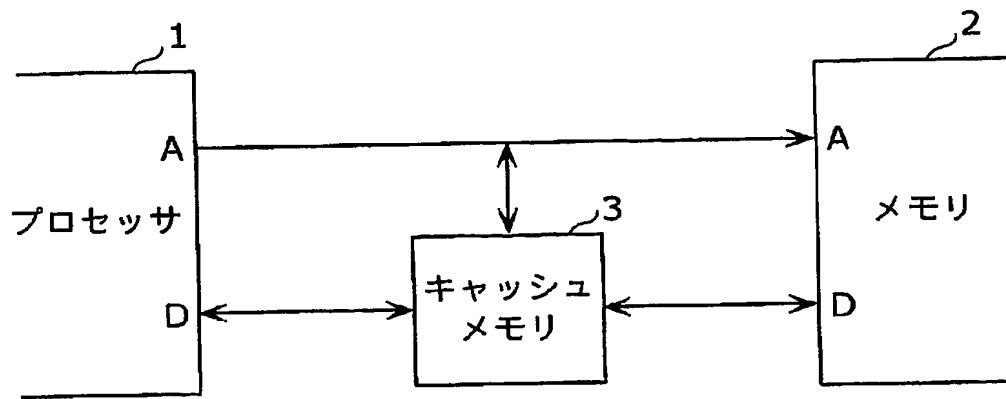
【符号の説明】

【0082】

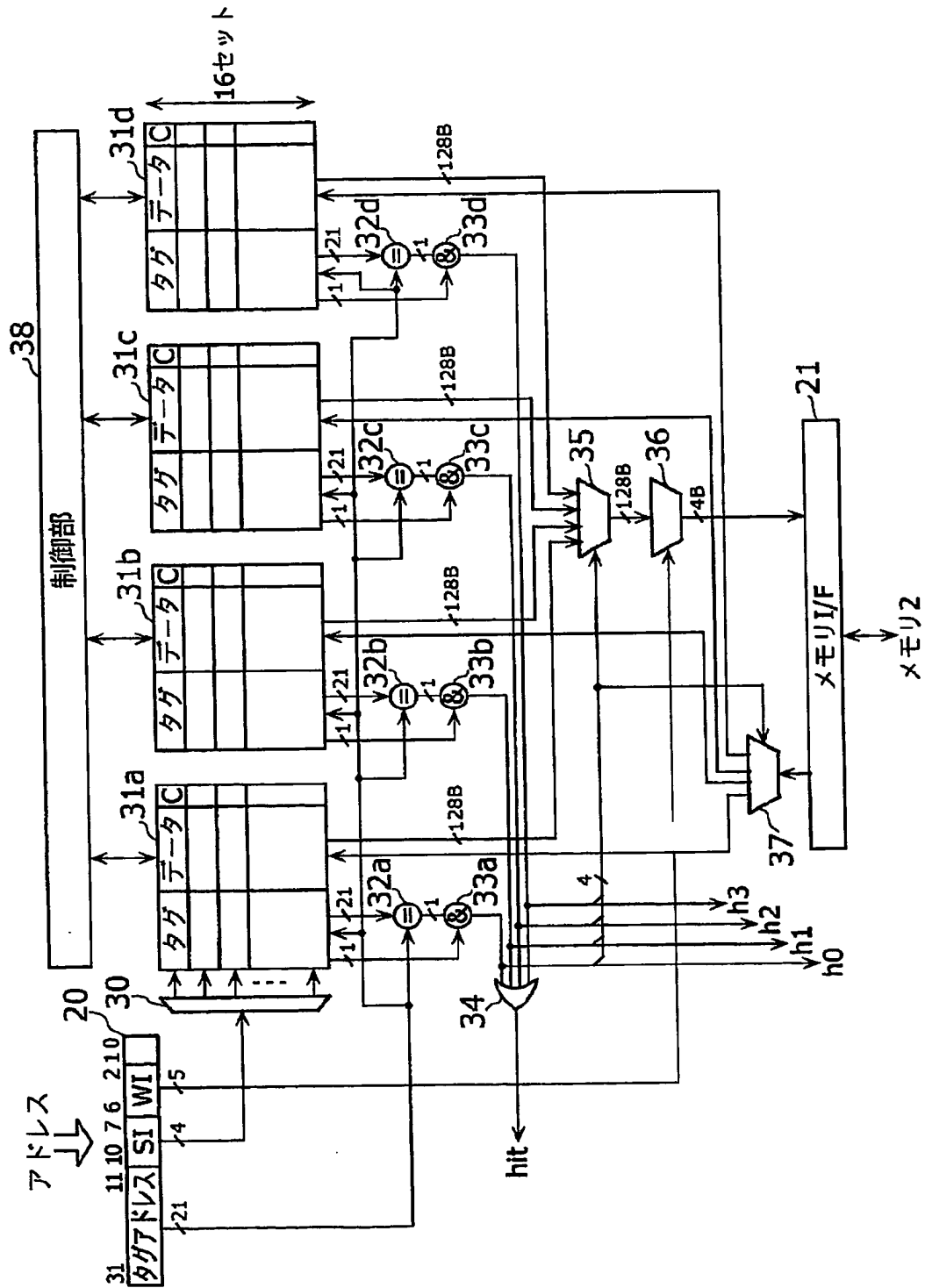
- 1 プロセッサ
- 2 メモリ
- 3 キャッシュメモリ
- 20 アドレスレジスタ
- 21 メモリ I/F
- 30 デコーダ
- 31 a ~ 31 d ウェイ
- 32 a ~ 32 d 比較器
- 33 a ~ 33 d アンド回路
- 34 オア回路
- 35 セレクタ
- 36 セレクタ
- 37 デマルチプレクサ
- 38 制御部
- 39 クリーニング処理部
- 40 Cフラグ設定部

4 1 リプレース部
1 3 1 b ~ 1 3 1 d ウェイ
1 3 8 制御部
1 3 9 クリーニング処理部
1 4 0 Wフラグ設定部
4 0 1 コマンドレジスタ
4 0 2 スタートアドレスレジスタ
4 0 3 サイズレジスタ
4 0 4 加算器
4 0 5 スタートアライナ
4 0 6 エンドアライナ
4 0 7 フラグ書換部

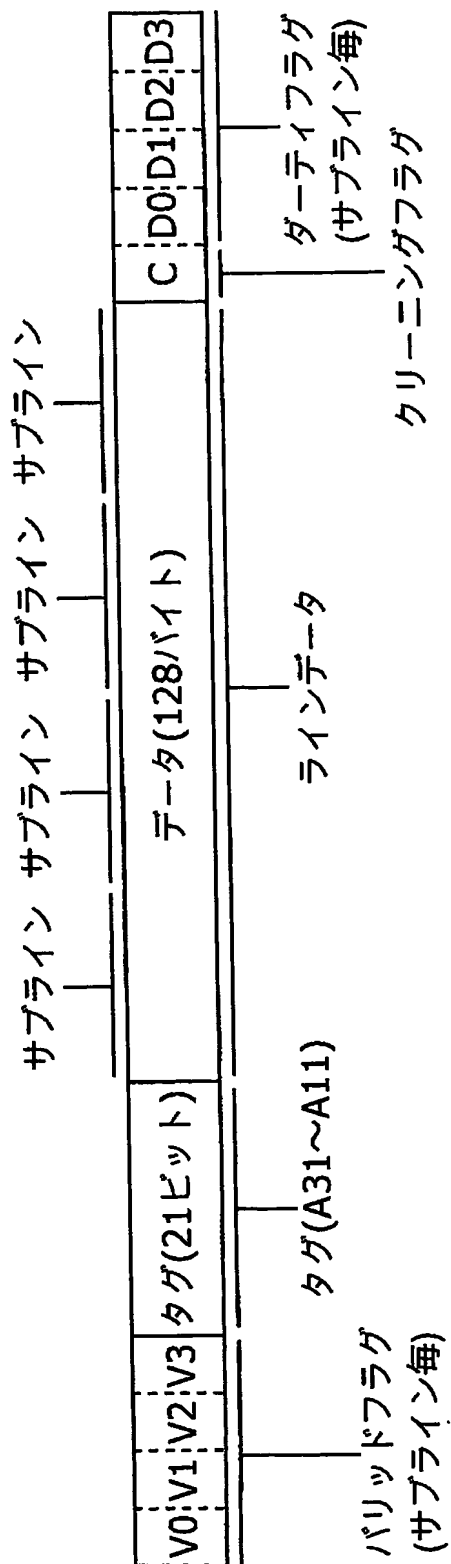
【書類名】図面
【図 1】



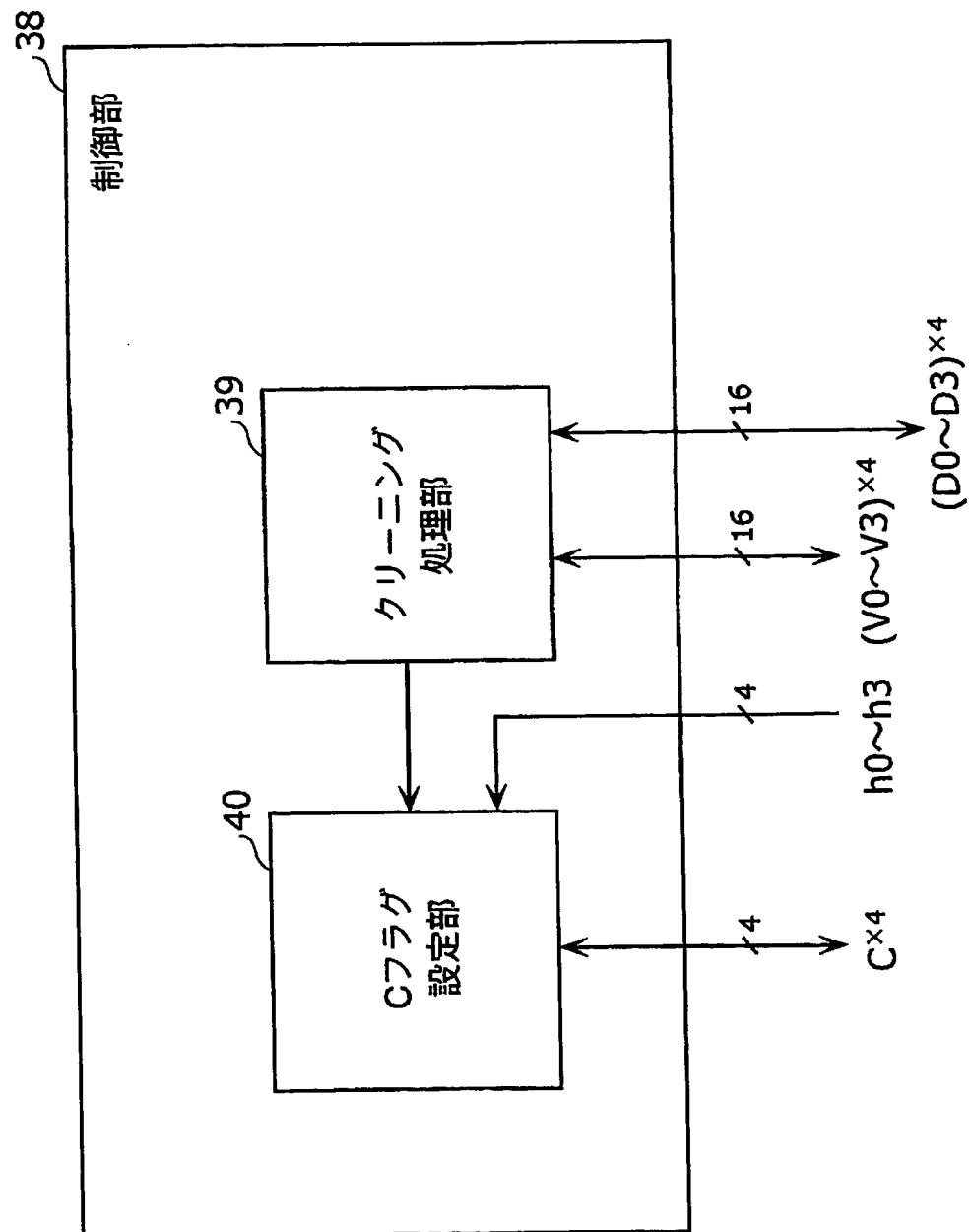
【図 2】



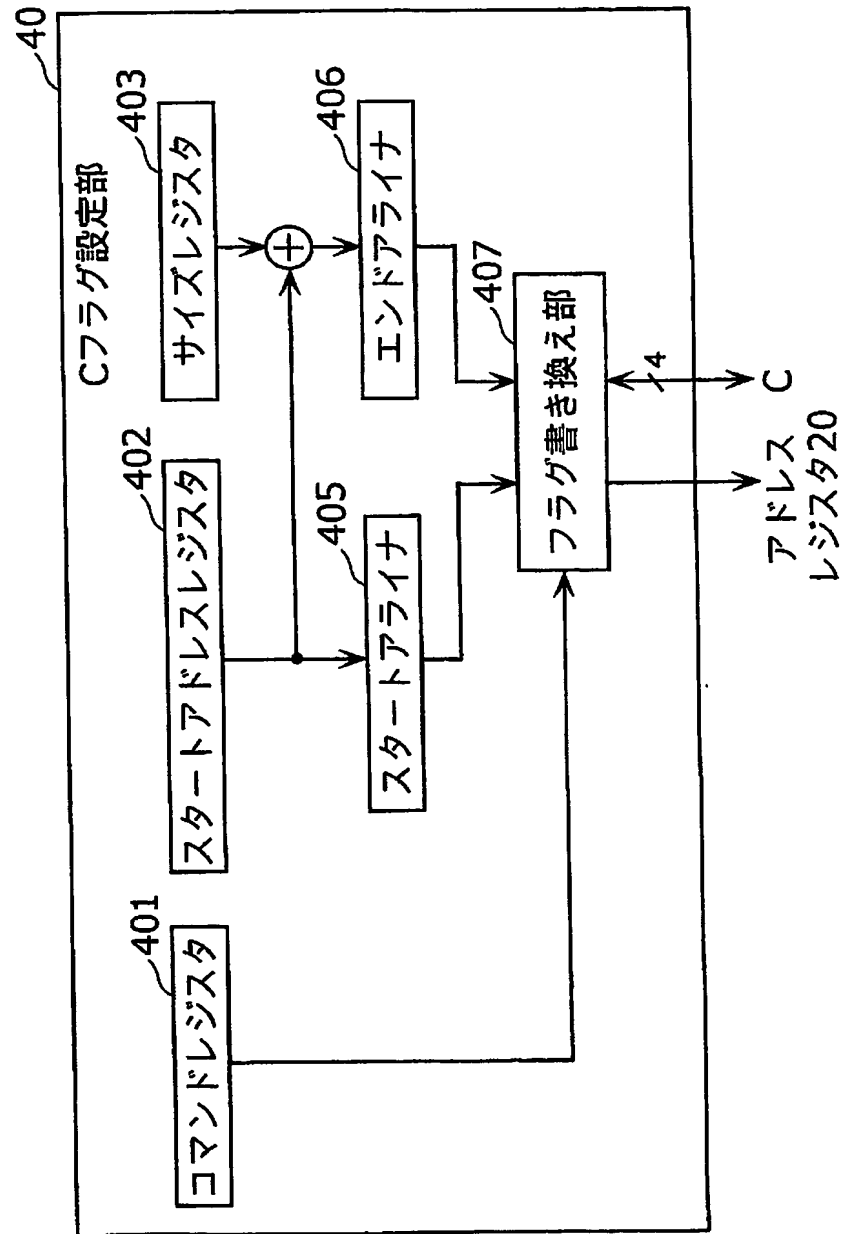
【図 3】



【図 4】



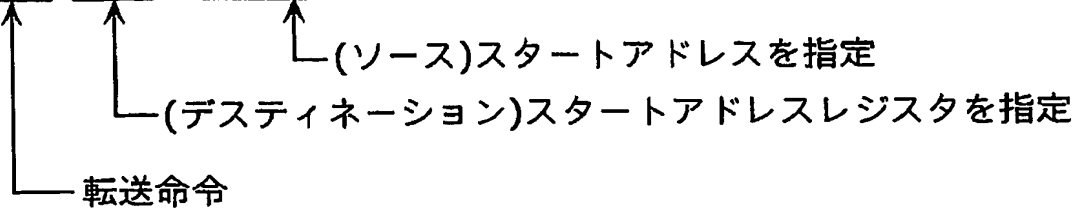
【図 5】



【図 6】

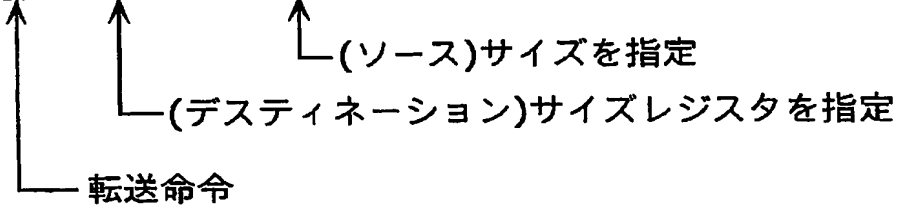
(a)

mov SAR , start_adrs



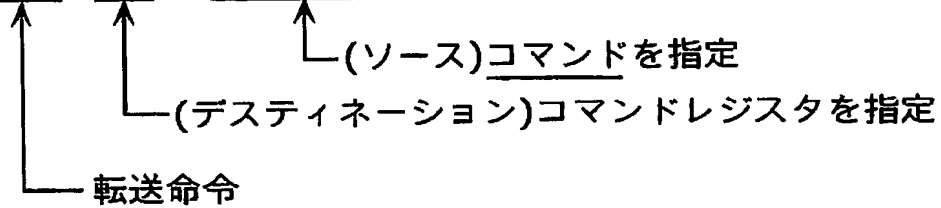
(b)

mov SR , size

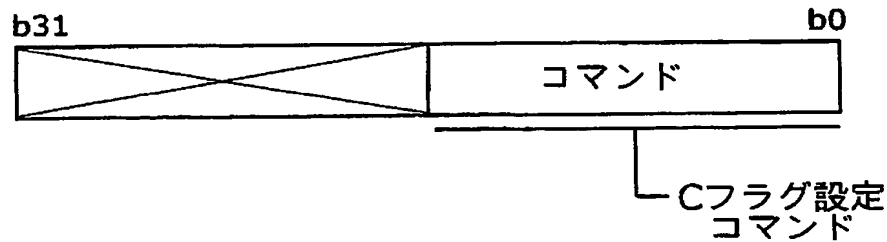


(c)

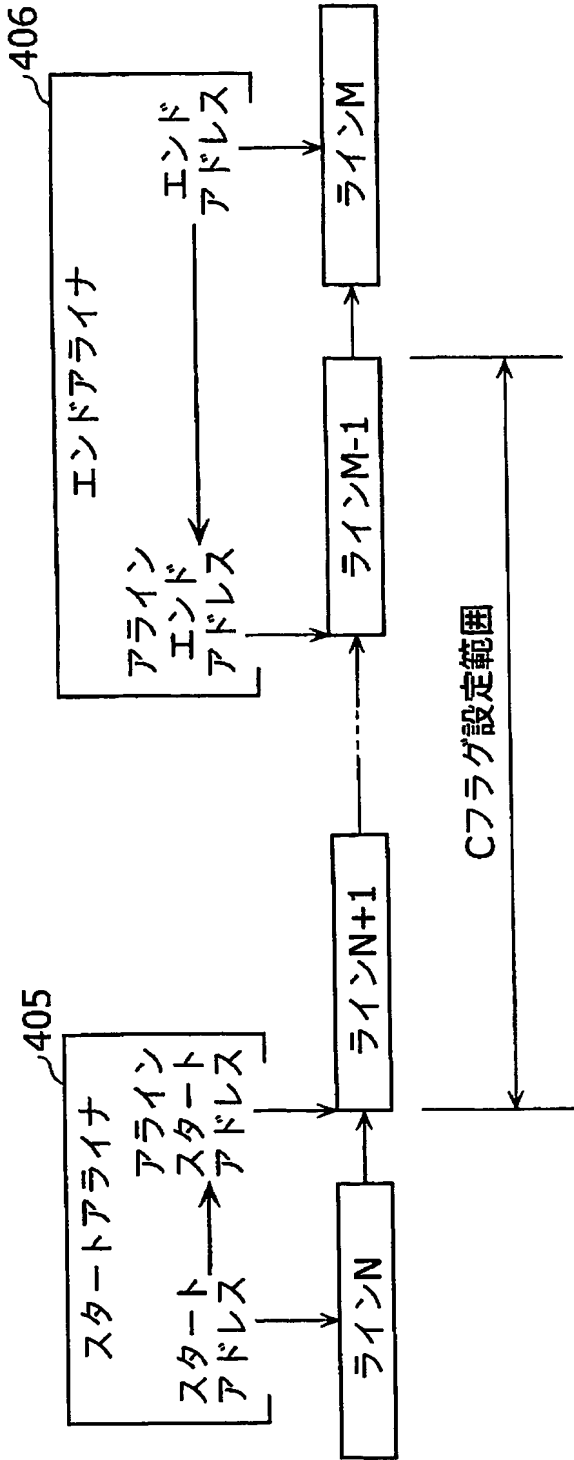
mov CR , command



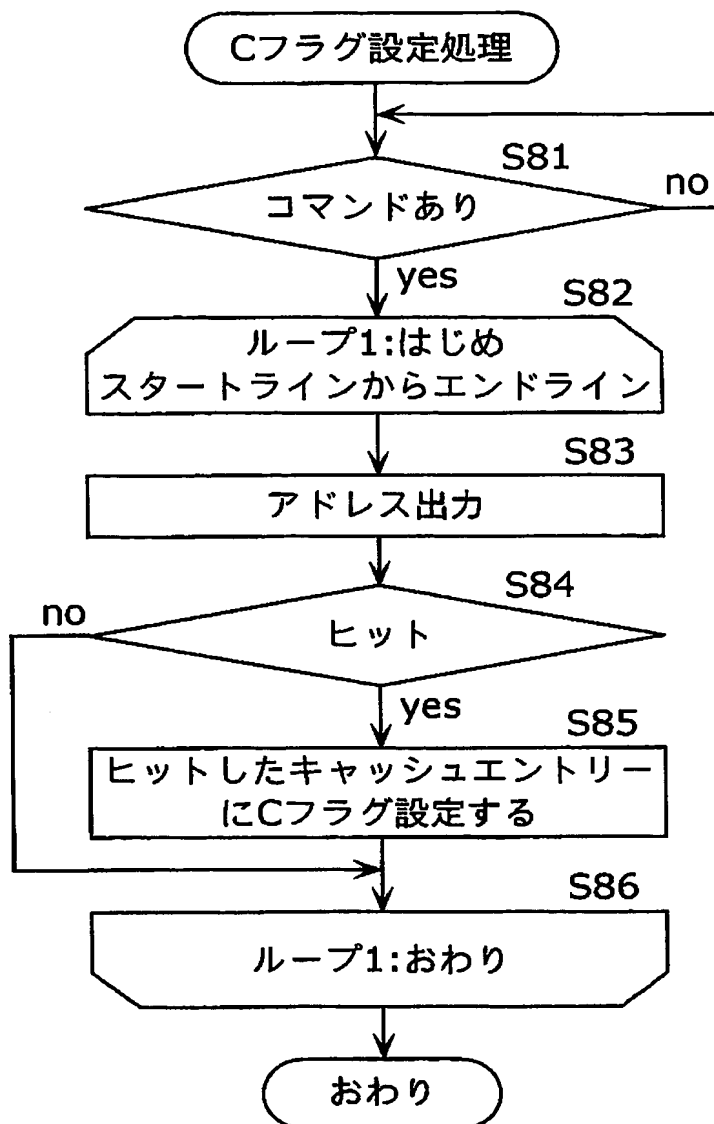
(d)



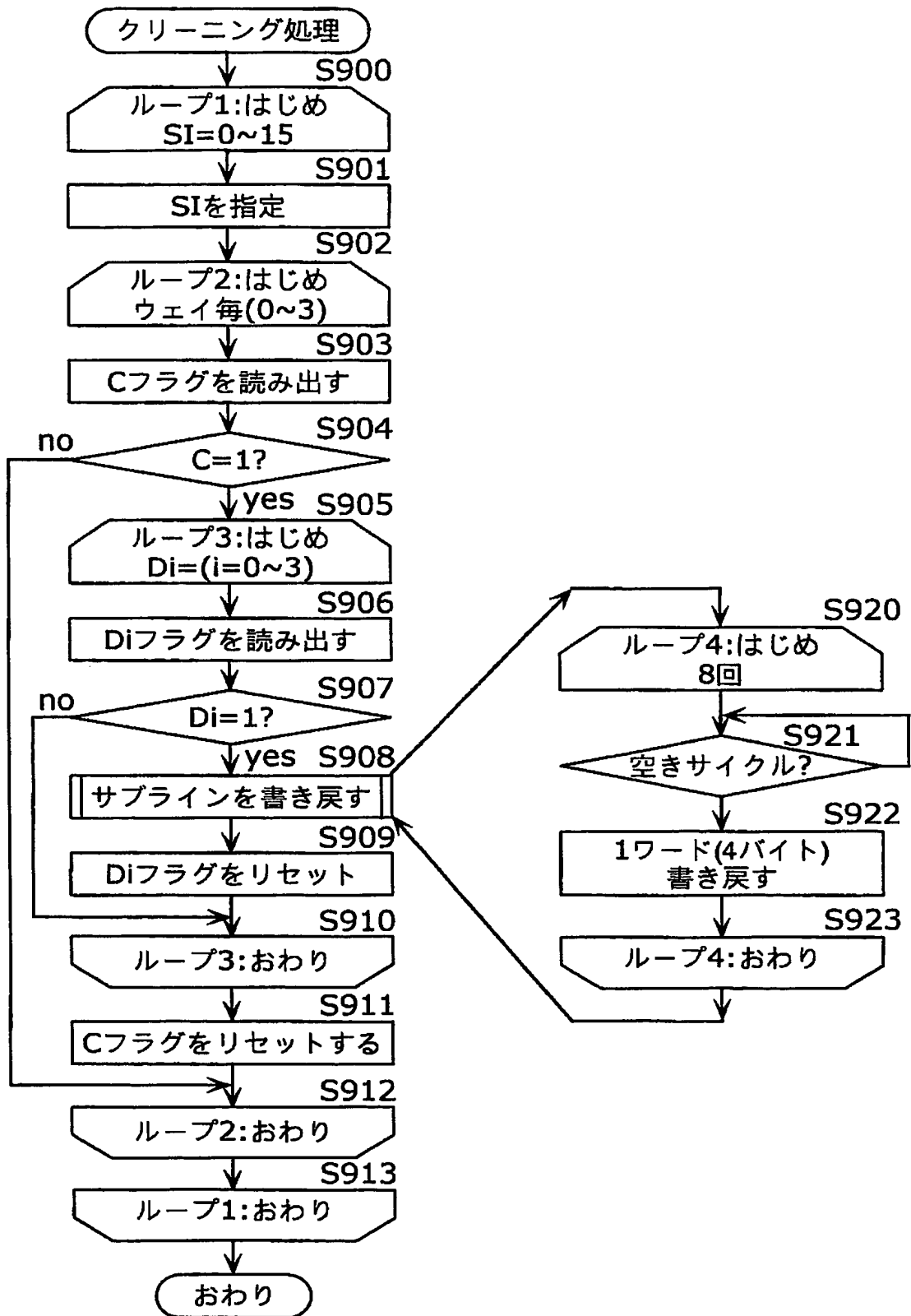
【図 7】



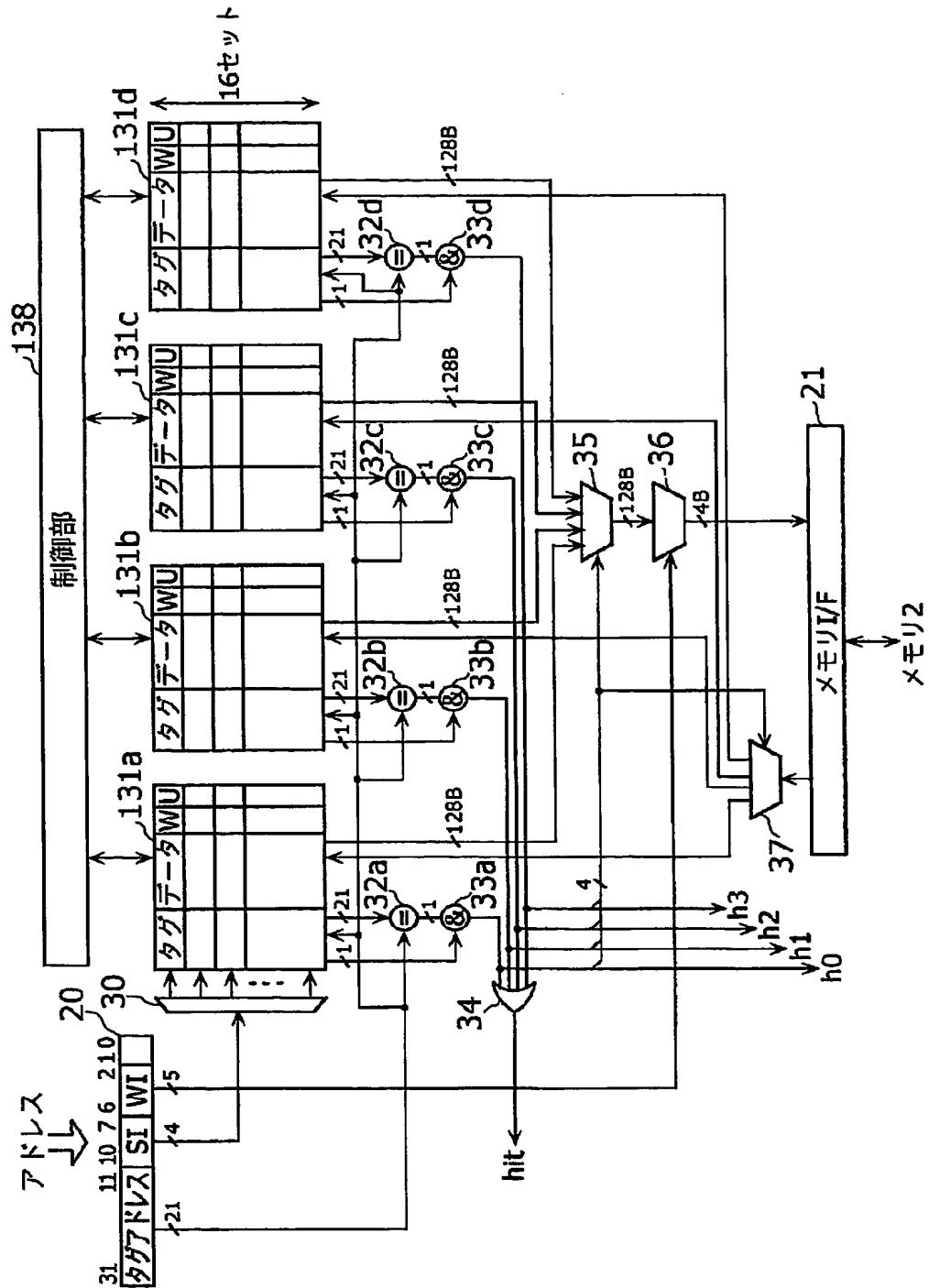
【図 8】



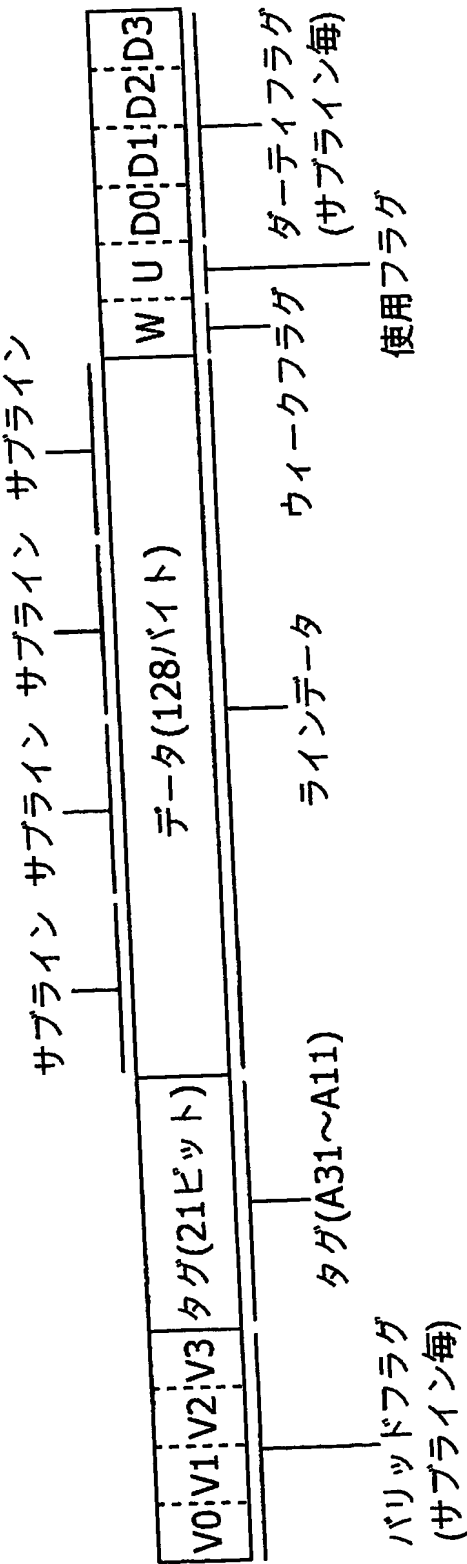
【図9】



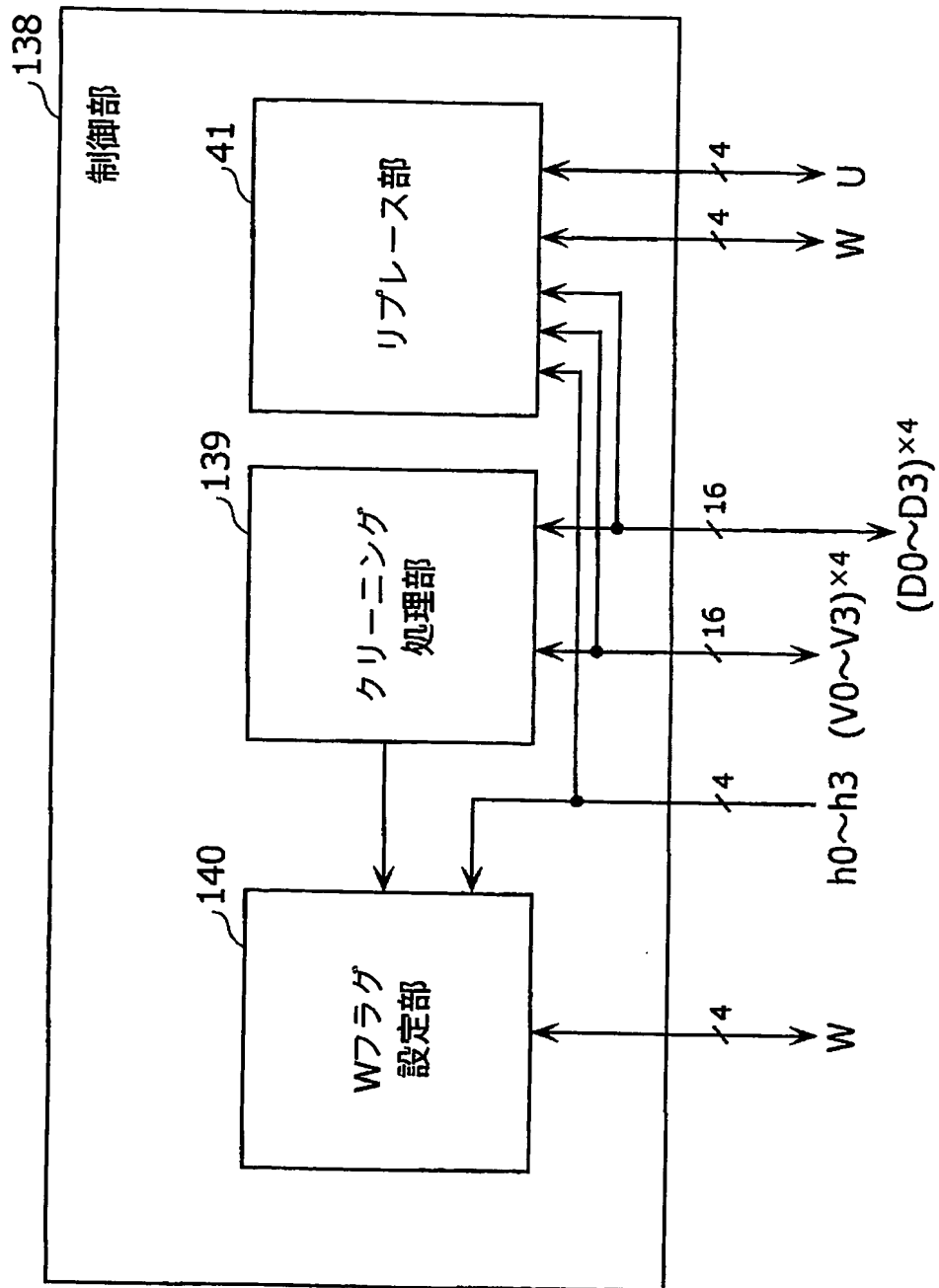
【図 10】



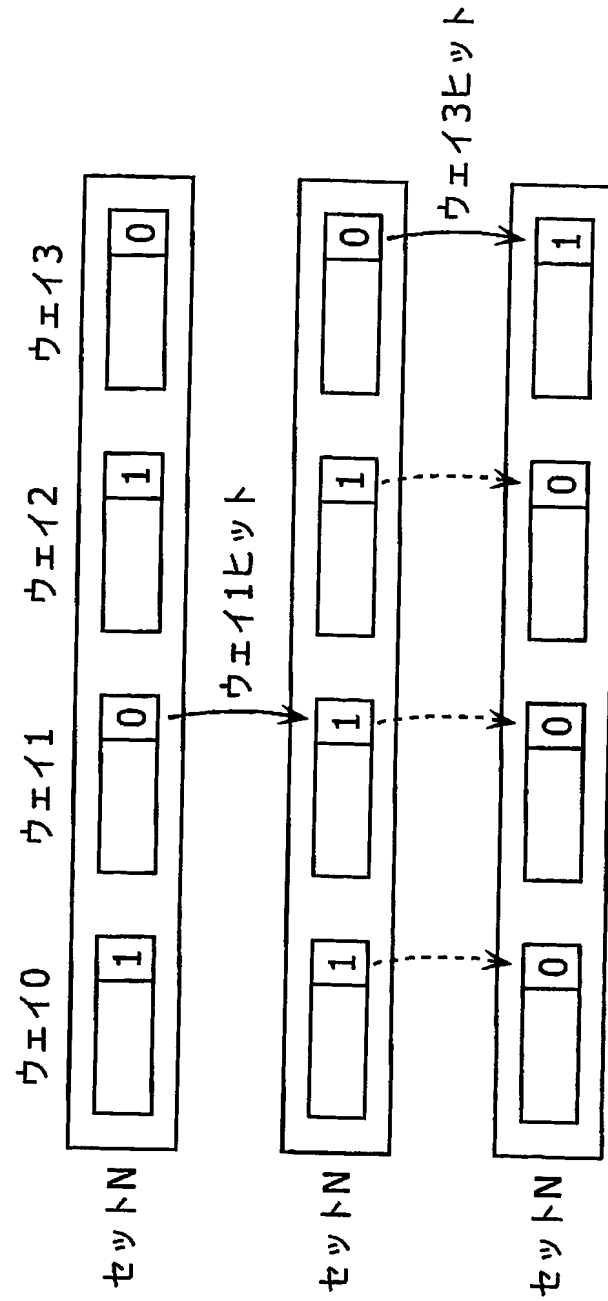
【図 11】



【図 12】

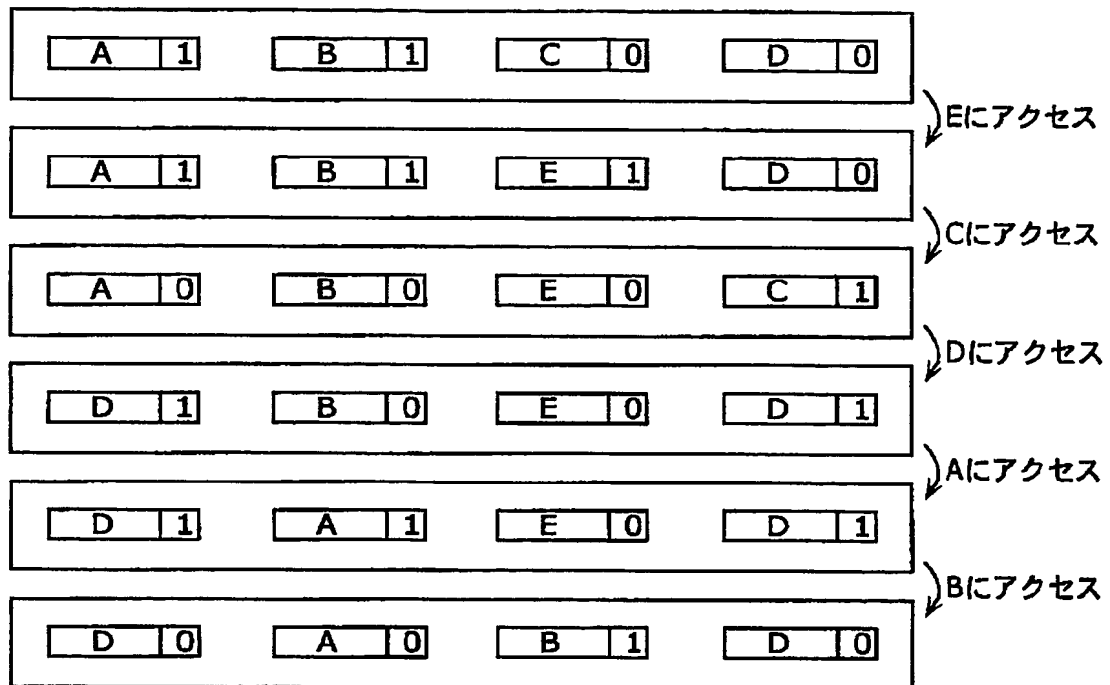


【図 13】

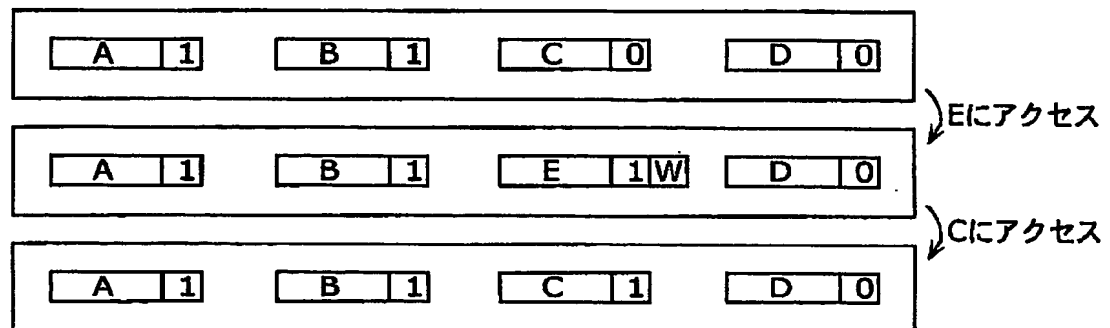


【図 14】

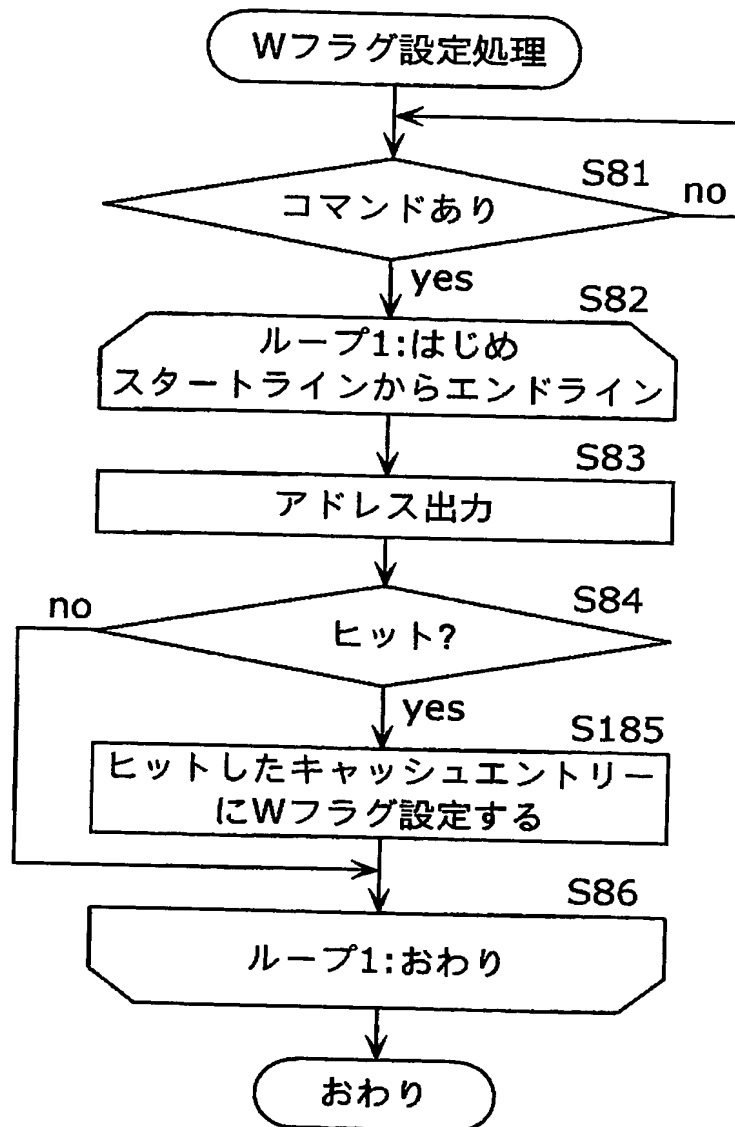
(a)



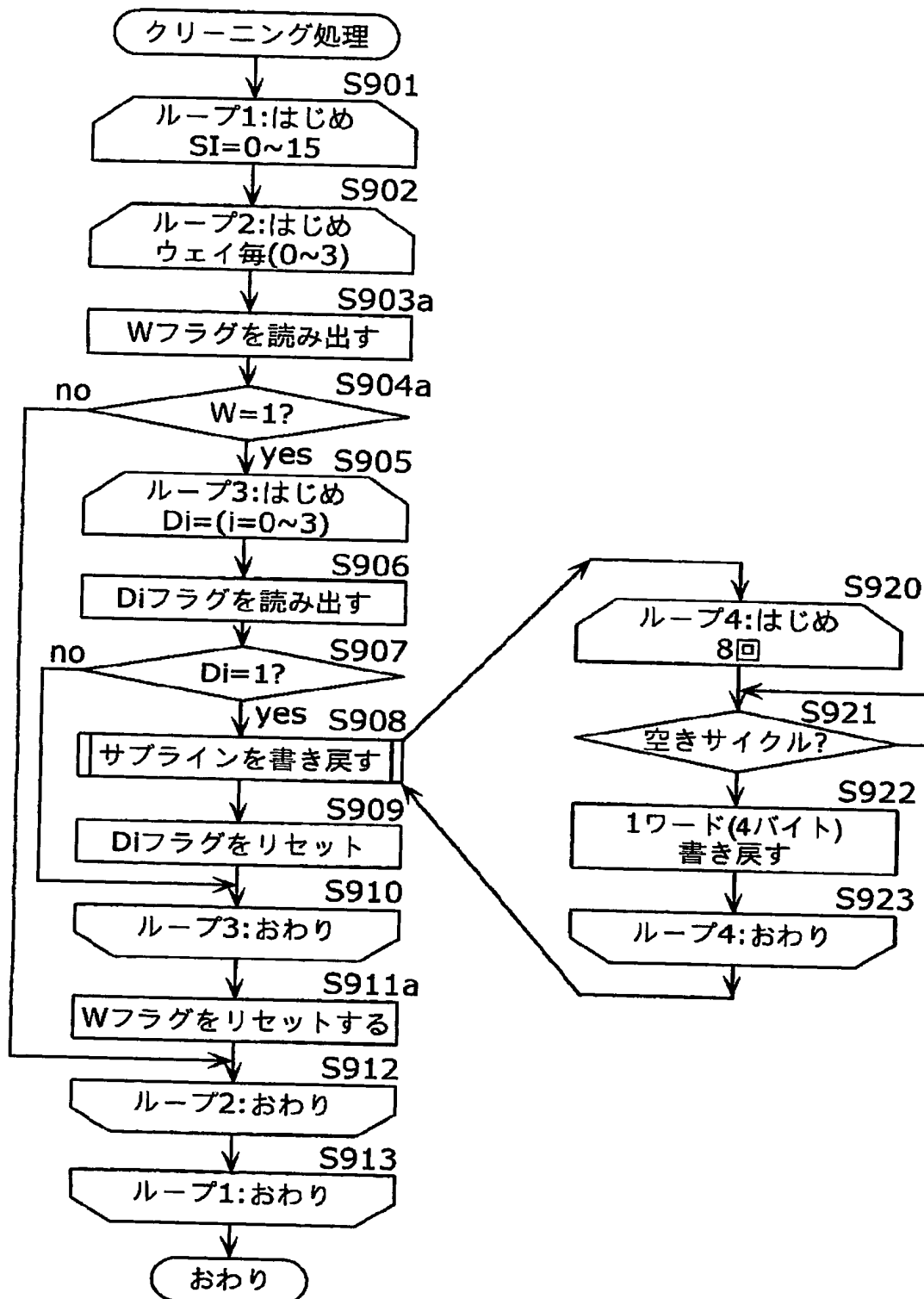
(b)



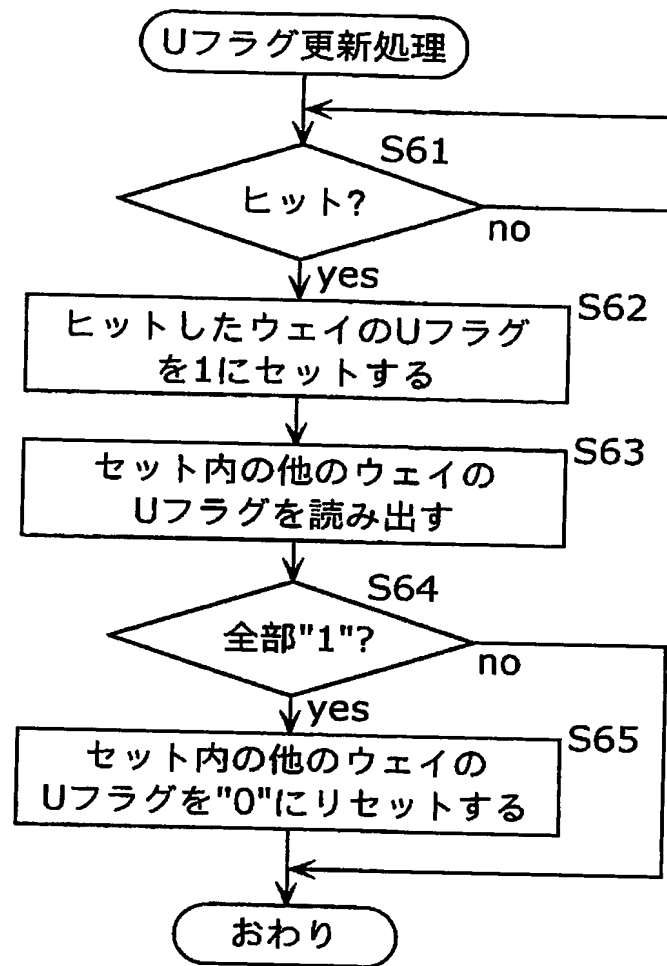
【図 15】



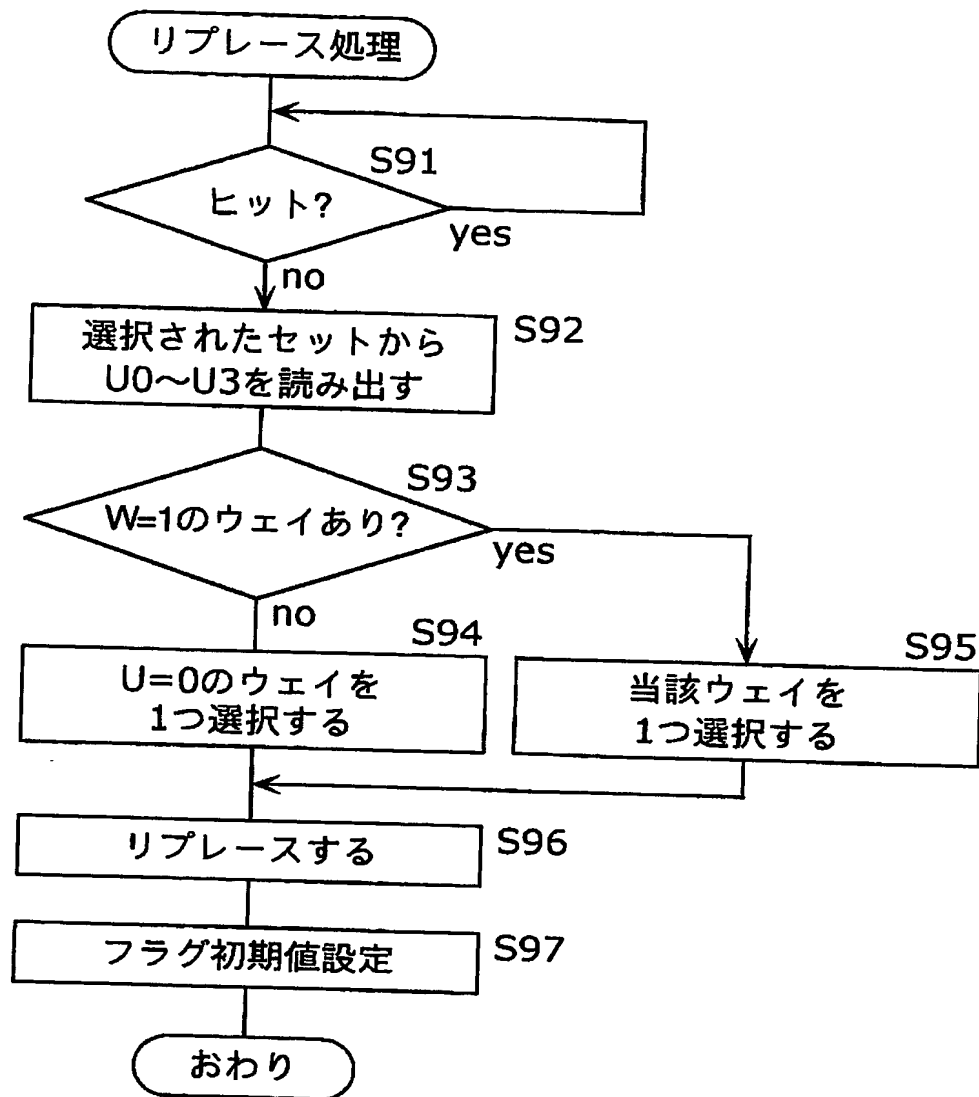
【図 16】




【図 17】



【図 18】





【書類名】要約書

【要約】

【課題】 キャッシュミス時のライトバックペナルティを効率よく低減するキャッシュメモリを提供する。

【解決手段】 ラインデータを保持するキャッシュエントリ毎に、当該キャッシュエントリに以降に書き込みが行われないかどうかを示すクリーニングフラグCを付与するCフラグ設定部40と、書き込みが行われないことを示すクリーニングフラグCが付与され、かつ書き込みされたことを示すダーティフラグDがセットされているキャッシュエントリのラインデータをメモリへライトバックするクリーニング処理部39とを備える。

【選択図】

図4



特願 2003-387350

ページ: 1/E

認定・付加情報

特許出願の番号	特願 2003-387350
受付番号	50301899679
書類名	特許願
担当官	第七担当上席 0096
作成日	平成15年11月19日

<認定情報・付加情報>

【提出日】 平成15年11月18日



特願 2003-387350

ページ: 1/E

出願人履歴情報

識別番号

[000005821]

1. 変更年月日
[変更理由]

住所
氏名

1990年 8月28日

新規登録

大阪府門真市大字門真1006番地
松下電器産業株式会社